Amendments to the Claims

1. (Currently Amended) A method for minimizing an amount of data needed to test a geometry chunk in a frame against subarea boundaries in a compositing window, comprising the steps of:

defining the geometry chunk with a bounding region, wherein said bounding region defines a space the geometry chunk occupies on the compositing window;

storing said bounding region for use in processing the geometry chunk in a subsequent frame;

sending said bounding region to graphics pipelines;

determining a graphics pipeline of said graphics pipelines that will render the geometry chunk defined by said bounding region;

assigning a subarea in the compositing window to receive an output of said graphics pipeline; and

communicating the geometry chunk to said graphics pipeline that will render the geometry ~~chunk.~~ chunk;

wherein said graphics pipelines are configured to render the frame by spatial compositing through parallel processing.

2. (Previously Presented) The method of claim 1, wherein the geometry chunk is comprised of a piece of a geometry provided by a graphics application.

3. (Previously Presented) The method of claim 1, wherein said space is a screen space.

4. (Previously Presented) The method of claim 1, wherein said space is a world space.

5. (Previously Presented) The method of claim 1, wherein said space is an object space.

6. (Canceled)

7. (Canceled)

8. (Original) The method of claim 1, wherein the geometry chunk is represented as a display list.

9. (Original) The method of claim 1, wherein the geometry chunk is represented as a vertex array object.

10. (Original) The method of claim 1, wherein the geometry chunk is represented as buffered vertices.

11. (Currently Amended) A system for minimizing an amount of data needed to test a geometry chunk in a frame against subarea boundaries in a compositing window, comprising:

graphics pipelines;

a geometry distributor that defines a bounding region for the geometry chunk, sends said bounding region to said graphics pipelines to determine a graphics pipeline of said

graphics pipelines that will render the geometry chunk defined by said bounding region, assigns a subarea in the compositing window to receive an output of said graphics pipeline, and communicates the geometry chunk to said graphics pipeline that will render the geometry chunk; and

a memory that stores said bounding region for use in processing the geometry chunk in a subsequent frame;

wherein said bounding region defines a space the geometry chunk occupies on the compositing ~~window.~~ window;

wherein said graphics pipelines are configured to render the frame by spatial compositing through parallel processing.

12. (Original) The system of claim 11, further comprising a graphics application that provides the geometry chunk to said geometry distributor.

13. (Original) The system of claim 12, wherein said geometry distributor comprises a virtual graphics unit that interfaces with said graphics application.

14. (Previously Presented) The system of claim 11, wherein the geometry chunk is comprised of a piece of a geometry provided by a graphics application.

15. (Previously Presented) The system of claim 11, wherein said space is a screen space.

16. (Previously Presented) The system of claim 11, wherein said space is a world space.

17. (Previously Presented) The system of claim 11, wherein said space is an object space.

18. (Previously Presented) The system of claim 11, wherein said geometry distributor comprises:

a bounding region calculator that calculates said bounding region for the geometry chunk;

a graphics pipeline assignor that assigns said graphics pipeline to said subarea in the compositing window; and

a graphics pipeline distributor that distributes the geometry chunk to the graphics pipeline.

19. (Original) The system of claim 11, wherein the geometry chunk is represented as a display list.

20. (Original) The system of claim 11, wherein the geometry chunk is represented as a vertex array object.

21. (Original) The system of claim 11, wherein the geometry chunk is represented as buffered vertices.

The listing of claims will replace all prior versions, and listings of claims in the application.